AMENDMENTS TO THE CLAIMS:

1. (Currently amended) A software method of improving at least one of efficiency and speed in executing a linear algebra subroutine on a computer having a floating point unit (FPU) with floating point registers (FRegs) and a load/store unit (LSU) capable of overlapping loading data and processing said data by the FPU, said FPU being interfaced with an L1 cache and having an L1 cache/FReg interface "loading penalty of *n* cycles", *n* being an integer greater than or equal to 1, during which data is rearranged in up to *n* cycles in said FRegs because data arrives out of order for said processing, said method comprising:

loading matrix data from a memory through a cache system at a fastest possible rate; and

then either immediately or at a later time, for an execution code controlling operation of said linear algebra subroutine execution, overlapping by preloading data into ~~floating point registers (~~ said FRegs ~~)~~of said FPU and then rearranging the data in said FRegs for up to said *n* cycles, said overlapping causing said matrix data to arrive into said FRegs from said L1 cache to be timely executed by the FPU operations of said linear algebra subroutine on said FPU.

2. (Previously presented) The method of claim 1, wherein instructions are unrolled repeatedly until the data loading reaches a steady state in which a data loading exceeds a data consumption.

3. (Original) The method of claim 1, wherein said linear algebra subroutine comprises a matrix multiplication operation.

4. (Previously presented) The method of claim 1, wherein said linear algebra subroutine comprises a subroutine equivalent to a LAPACK (Linear Algebra PACKage) subroutine.

5. (Original) The method of claim 4, wherein said LAPACK subroutine comprises a BLAS Level 3 L1 cache kernel.

6. (Currently amended) An apparatus, comprising:

a memory to store matrix data to be used for processing in a linear algebra program;

an L1 cache to receive data from said memory;

a floating point unit (FPU) to perform said processing; and

a load/store unit (LSU) to load data to be processed by said FPU, said LSU loading said data into a plurality of floating point registers (FRegs),

wherein said data processing overlaps said data loading such that matrix data is preloaded into said FRegs from said L1 cache prior to being required by said FPU and the preloaded data in said FRegs is rearranged for up to *n cycles, n* being an integer greater than or equal to 1.

7. (Original) The apparatus of claim 6, wherein said preloading is achieved by unrolling a loading instruction so that a load occurs every cycle until a preload condition has been satisfied.

8. (Original) The apparatus of claim 6, wherein said linear algebra program comprises a matrix multiplication operation.

9. (Previously presented) The apparatus of claim 6, wherein said linear algebra program comprises a subroutine equivalent to a LAPACK (Linear Algebra PACKage) subroutine.

10. (Previously presented) The apparatus of claim 9, wherein said subroutine comprises a BLAS Level 3 L1 cache kernel.

11. (Original) The apparatus of claim 6, further comprising:

a compiler to generate an instruction for said preloading.

12. (Currently amended) A computer-readable storage medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform a method of improving at least one of speed and efficiency in executing a linear algebra subroutine on a computer having a floating point unit (FPU) and a load/store unit (LSU) capable of overlapping loading data and processing said data, said method comprising:

for an execution code controlling operation of said linear algebra subroutine execution, overlapping by preloading data into ~~a~~ floating point registers (FRegs) of said FPU and rearranging the preloaded data in said FRegs for up to *n* cycles, where *n* is an integer greater than or equal to 1, said overlapping causing data from an L1 cache to arrive into said FRegs, to be timely executed by FPU operations of said linear algebra subroutine on said FPU in view of said up to *n* cycles used for rearranging said preloaded data.

13. (Currently amended) The computer-readable storage medium of claim 12, wherein a load instruction is unrolled repeatedly until the data loading reaches a steady state in which a data loading exceeds a data consumption.

14. (Currently amended) The computer-readable storage medium of claim 12, wherein said linear algebra program comprises a matrix multiplication operation.

15. (Currently amended) The computer-readable storage medium of claim 12, wherein said linear algebra program comprises a subroutine equivalent to a LAPACK (Linear Algebra PACKage) subroutine.

16. (Currently amended) The computer-readable storage medium of claim 15, wherein said subroutine comprises a BLAS Level 3 L1 cache kernel.

17. (Currently amended) A method of providing a service involving at least one of solving and applying a scientific/engineering problem, said method comprising at least one of:

using a linear algebra software package that computes one or more matrix subroutines, wherein said linear algebra software package generates an execution code controlling a load/store unit loading data into ~~a~~ floating point registers (FRegs) for a floating point unit (FPU) performing a linear algebra subroutine execution, said FPU capable of overlapping loading data and performing said linear algebra subroutine processing, such that, for an execution code controlling operation of said FPU, said overlapping causes a preloading of data from an L1 cache into said FRegs and then rearranges said preloaded data for up to *n* cycles, *n* being an integer greater than or equal to 1, and wherein a stride one data transfer is used for providing said data for said preloading for all operands without using a data copy processing for correcting said stride one data transfer for any operand of said linear algebra subroutine;

providing a consultation for purpose of solving a scientific/engineering problem using said linear algebra software package;

transmitting a result of said linear algebra software package on at least one of a network, a signal-bearing medium containing machine-readable data representing said result, and a printed version representing said result; and

receiving a result of said linear algebra software package on at least one of a network, a signal-bearing medium containing machine-readable data representing said result, and a printed version representing said result.

18. (Previously presented) The method of claim 17, wherein said linear algebra subroutine comprises a subroutine equivalent to a LAPACK (Linear Algebra PACKage) subroutine.

19. (Previously presented) The method of claim 18, wherein said subroutine comprises a BLAS Level 3 L1 cache kernel.

20. (Canceled)

21. (New) The method of claim 1, wherein said fastest possible rate comprises transferring said matrix data for said processing from said memory to said cache system in a stride one format, said method further comprising:

providing six kernels to be selectively available for said linear algebra subroutine execution, said selectively available six kernels thereby permitting said stride one transfer to be made without using a data copy processing for correcting said matrix data for any operand of said linear algebra subroutine.